SRI International

AN ICONIC TRANSFORM FOR SKETCH COMPLETION AND SHAPE ABSTRACTION

Technical Note 195

October 1979

By: Martin A. Fischler, Senior Computer Scientist
Phyllis Barrett, Programmer

Artificial Intelligence Center
Computer Science and Technology Division

SRI Project 5300

The work reported herein was supported by the National Science Foundation under Grant No. ENG-76-84623, and by the Defense Advanced Research Projects Agency under Contract No. DAAG29-76-C-0057.

SRI International®

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
OCT 1979		**00-10-1979 to 00-10-1979**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
An Ionic Transform for Sketch Completion and Shape Abstraction	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **SRI International,333 Ravenswood Avenue,Menlo Park,CA,94025**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**43**	

AN ICONIC TRANSFORM FOR SKETCH COMPLETION AND SHAPE ABSTRACTION*

By: M. A. Fischler and P. Barrett
SRI International
Menlo Park, CA 94025

I ABSTRACT

This paper shows how a simple label propagation technique, in conjunction with some novel ideas about how labels can be applied to an image to express semantic knowledge, lead to the simplification of a number of diverse and difficult image analysis tasks (e.g., sketch completion and shape abstraction). A single algorithmic technique, based on skeleton and distance transform concepts, is applied to appropriately labeled images to obtain the desired results. A key point is that the initial semantic labeling is not required at every location in the image, but only at those few critical locations where significant changes or discontinuities occur.

II INTRODUCTION

A description of a scene can employ two distinct types of representations--iconic and symbolic. In an iconic representation derived image structures are substituted for portions of the original image; these derived structures are generally based on transformations that emphasize, or makes explicit, selected aspects of the scene. In a symbolic representation, the names of reference structures (i.e., predefined models) are assigned to coherent segments of the image.

---

* This work was supported by the National Science Foundation under Grant ENG 76-84623, and by the Defense Advanced Research Projects Agency under Contract No. DAAG29-76-C-0057.

Both types of representations are useful. Because all possible image structures cannot be modeled, iconic representations are needed for completeness; they also are employed in the early stages of processing required by most scene analysis paradigms, and are often adequate in themselves for such tasks as image partitioning and image matching. Symbolic representations, when applicable, can include information about the scene that may not be apparent from the immediate image context.

Typical examples of iconic transformations are convolution and thresholding. For example, if an image is convolved with a high pass filter or edge detector and then thresholded, an iconic representation of the scene, emphasizing the lines, edges, and intensity discontinuities in the scene is obtained. For the purpose of partitioning the image into discrete objects, or for measuring the sizes of objects, this representation might be a more appropriate description of the scene than the original image.

With few exceptions (e.g., Hanson and Riseman [1978], Rosenfeld et al. [1976], Tenenbaum and Barrow [1976], and Waltz [1978]), the existing concept of the role of an iconic transform is to operate on an image devoid of explicit semantic information, either to achieve some final goal, or to "preprocess" the image so as to simplify the task of extracting a symbolic description. The symbolic description is either the desired end product, or is the basis for further symbolic processing. Thus, the conventional scene analysis paradigm sharply partitions the iconic and symbolic aspects of the image analysis task; the iconic aspect is generic and keyed to enhancing selected attributes of general image content, while the symbolic aspect is much more context-sensitive and goal-directed.

In this paper, we show how symbolic/semantic information can directly interact with an iconic process (the Labeled Distance Transform) so as to achieve a goal-oriented transformation of an image without the need to define specific object models, or to first derive a complete symbolic description of the scene.

The main contribution of this paper is the introduction of a new tool for scene analysis called the Labeled Distance Transform (LDT). Because the LDT is related to both distance transforms and skeletons, we discuss these topics before describing the LDT itself; we then describe how the LDT can be employed for both sketch completion and shape abstraction, and give examples of both of these applications; finally, we mention some additional applications and possible extensions of our current work.

III DISTANCE TRANSFORMS AND SKELETONS

At some stage in the analysis of many types of imagery, the image is represented as a 2-dimensional array of 1's and 0's, in which clusters of 1's correspond to meaningful components of the scene. The following are examples of the types of information we might wish to extract from such a representation:

(1) We might be uninterested in the absolute dimensions of a given cluster or object, but would like a simple description of its shape (as in the case of character recognition).

(2) We might want to measure some dimensional attribute of an isolated object (such as its minimum or maximum width as in industrial inspection problems concerned with flaw detection).

(3) We might want a description of the spatial relationships between the objects depicted in an image (for example, to verify that some manufactured assembly has all of its required pieces in their correct locations).

(4) We might want to compare images of the same object taken at sequential points in time, and characterize the nature of the shape changes that have occurred (for example, as in the study of the growth of living organisms).

The process of extracting the above information from a binary image has been greatly simplified by two computational techniques: the Medial Axis Transform (MAT) introduced by Blum [1967], and the Distance Transform (DT) introduced by Rosenfeld and Pfaltz [1966].

A. The Medial Axis Transform and Continuous Skeletons

The MAT produces a stick figure, or "skeleton" to represent the general shape of an object (or region) in a binary image. It derives this skeleton by marking those points interior to the object where any of the following equivalent conditions are satisfied:

(1) The centers of maximal disks (ones that fit inside the object, but are not contained in other disks also interior to the object) are the skeletal points.

(2) Consider the (minimum) distance from each point of the object to the object contour; those points that are minimally distant from more than one contour point are the skeletal points.

(3) Imagine an object whose border is set on fire; if the fire spreads in all directions with equal velocity, then the points at which it quenches itself are the skeletal points ("grassfire" definition).

(4) Those points at which the derivatives of distance or direction (from the nearest boundary point) have a discontinuity are the skeletal points.

Important properties of the skeleton produced by the MAT are:

(1) The contours of the object can be exactly reconstructed from its skeleton.

(2) The geometry of the skeleton is invariant under image translation or rotation.

(3) The skeleton is connected.

Blum [1967], Blum and Nagel [1978], Bookstein [1978, 1979], and Calabi and Hartnett [1968], have investigated the theory of skeletons and shape representation in continuous spaces (especially for continuous shapes on the 2-D Euclidean plane). However, practical computer techniques for representation and computation are restricted to discrete spaces and thus to approximations to the continuous theory. (For the case of shapes that can be approximated by straight line segments, both Montanari [1969] and Bookstein [1979] have described computationally complex but feasible techniques for generating continuous skeletons.)

The (discrete) distance transform techniques developed by Montanari [1968], Quam [1979], and Rosenfeld and Pfaltz [1966, 1967], provide a

practical means for computing a discrete approximation to the continuous skeleton, but the MAT properties listed above are no longer strictly assured. There is the further problem that the noisy, quantized, and viewpoint-dependent imagery produced by physical imaging devices often leads to skeletons which have branches that are deemed artifacts; these branches must be pruned to obtain a skeleton with acceptable topology. The pruning operation almost invariably involves the need to introduce ad-hoc thresholds with all of their well known deficiencies.

B. The Distance Transform and Discrete Skeletons

Rosenfeld and Pfaltz [1966] showed that if we are willing to use a "city-block" distance metric [in which the distance between two points (X1,Y1) and (X2,Y2) is $|X1-X2|+|Y1-Y2|$] rather than a Euclidean metric, then the image distance of every point in an object from its nearest boundary point can be determined by adjusting the "distance" assigned to each such point based on the value of two of its immediate neighbors during two sequential passes through the image. The DT can be used both to obtain the skeleton and to answer questions about the width of the corresponding object.

Montanari [1968] presented a variation of Rosenfeld's DT that can provide values approximating a Euclidean metric (accurate to within some specified percentage of the true distance), but at a much higher computational cost.

In some recent work, Quam [1979] has shown that if pointers (vectors) to the nearest boundary points are retained during the computation of the DT, then with the equivalent of a 4-pass algorithm we can obtain a distance transform (which we call the "Vector Distance Transform," or VDT) that provides Euclidean distances in the image conjectured to be accurate to within .125 pixels of the exact answer. He further showed that if a "pruned" skeleton is generated based on discontinuities in the directions associated with the vectors, rather than in the derivatives of the distance (as is typically done when the directional information is not available), then a more intuitively

satisfying skeleton can usually be generated. In this paper, we extend the concept of the distance transform in two distinct ways. First, we allow "semantic labels" to be assigned to subsets of pixels in the input image; these semantic labels propagate along with the distance from the "closest" originating pixel. Typically, the same semantic label is associated with a contiguous subset of the (boundary) points of an object or region. Next, we allow specially labeled pixels in the input image to alter or block the normal distance propagation update mechanism (e.g., distances do not change uniformly as we pass over an occlusion edge in an image of a 3-D scene).

We show that these modifications permit us to generate a pruned skeleton without the necessity of assigning arbitrary "discontinuity" thresholds to either distance or direction; they actually allow us to control the shape of the resulting skeleton and avoid "false" branches due to minor perturbations in the shape of the boundary, and thus produce a skeleton that is semantically more meaningful than any of the previously mentioned techniques.

We also show how the use of a non-binary label-based distance transform can be used to sketch (or smooth) the missing parts of boundaries and regions with applications to more general scene analysis tasks.

Finally, we indicate how a continuous skeleton can be generated (from a finite point set) in a computationally effective manner by the appropriate use of labeling techniques.

IV THE LABELED DISTANCE TRANSFORM

The city-block metric Distance Transform (DT) operates on a binary image to produce an integer-valued array. For conceptual purposes we assume that the initial binary values are zero and #, (where # is a large number). The final value assigned to each pixel location gives the city-block distance from the nearest location in the original array

that had a zero label; the zero-labeled points never have their values altered.

The DT algorithm requires a left-to-right/top-to-bottom pass through the array, followed by a right-to-left/bottom-to-top pass. During the first pass, the integer value assigned to each location is the minimum of {its original value, one plus the value at the location immediately to its left, or one plus the value at the location immediately above it}. During the second pass, the integer value assigned to each location is the minimum of {its current value, one plus the value at the location immediately to its right, or one plus the value at the location immediately below it}.

Conceptually, the LDT assumes that a label will be associated with each pixel location in addition to the distance value discussed above. Initially, each pixel location with a zero distance value will also have an identifying label; the non-zero points will initially have null labels. When the algorithm described above updates the distance value at a given pixel location, the label of the neighboring cell providing the minimum distance replaces the label at the given location.

If we assign each zero point a unique label (e.g., its X,Y coordinate values), we have described the equivalent of the VDT for the city-block distance metric. More typically, we assign a common label to all the zero points of an isolated object (for sketch completion or scene partitioning); or assign a common label to all the zero points along some segment of the boundary of an object whose interior consists of non-zero points (for shape abstraction / skeleton generation).

The basic LDT algorithm is presented in Appendix A. In addition to the normal distance and label propagation procedures described above, there is also the procedure for altering or blocking propagation (discussed in the section on sketch completion), and the procedure for producing a skeleton (discussed in the section on shape abstraction).

V SHAPE ABSTRACTION (generating a controlled internal skeleton)

The skeleton, or medial axis, makes explicit the shape information implied by the external boundary of an imaged object. One main deficiency of existing algorithms for generating the internal skeleton of an object is that minor perturbations of the boundary due to noise, or artifacts of the imaging process, or even due to idiosyncrasies of the particular instance of the shape under consideration, can lead to a skeleton whose topology does not reflect those attributes of shape that are relevant to its intended use.

A key point here is that any abstraction must be based on (possibly implicit) assumptions indicating which shape attributes of the original object are important, and thus are to be preserved in some form in the final product. Since different intended uses of the abstraction can be associated with different aspects of the original object, it is desirable to be able to specify those shape attributes of the boundary we wish to influence the topology of the resulting skeleton; the LDT provides a simple mechanism for achieving this goal.

The primary mechanism for introducing semantic constraints on the topology of a skeletal abstraction via the LDT is to allow some external mechanism to identify those points on the boundary of the original object that are significant shape landmarks rather than artifacts. Once these "critical points" have been selected, they are assigned unique labels (numbers), and all adjacent points on the boundary, in a counterclockwise direction until the next critical point is reached, are assigned the same label.

Figure 1 shows an example of the labeling process and the resulting skeleton; Figure 2 shows some additional examples of LDT-produced skeletons. It should be noted that the critical points do not have to coincide with locally distinguished points on the boundary of the original object, but rather can reflect more global shape properties. The critical points are not always explicitly marked in the figures

since it is shown in Appendix B that the vertices of order one of the LDT skeleton always lie on critical points. (In fact, they are in one-to-one correspondence with the critical points when there are at least two such points; the requirement for at least two critical points exists because a single critical point does not result in differential labeling of the contour.)

The operation of the basic version of the LDT can be viewed as the process of uniformly "growing" the labeled regions into the unlabeled portions of the image; each pixel ultimately receives a label identical to that of the closest labeled point in the input image. The skeleton traces the boundary between the differently labeled regions after the growth process has been completed, and can be computed during the backward (second) pass of the LDT algorithm. In Appendix A, the labeling of the skeletal points is shown as occurring in a separate (third) pass to simplify the description of the algorithm.

In addition to the internal skeleton, there is also an external portion which is discussed later. This external portion is deleted in the examples associated with this section of the paper by ANDing the skeleton with a binary mask of the original object.

Figure 3 compares the skeletons produced by the LDT with those produced by the formal definition on a continuous space (using an example taken from Bookstein [1979]). Key features of the comparison are (see Appendix B):

(1) The discrete LDT skeleton, with appropriately chosen critical points, subsumes the continuous skeleton (to within an accuracy of one pixel--some error is unavoidable when representing a continuous structure on a quantized space); it extends the continuous skeleton out to the object boundary at the critical point locations (see Figure 3a,b,c).

(2) If a critical point used in the generation of an LDT skeleton is deleted, then the corresponding branch of the skeleton is not produced; the remaining skeleton geometry is not altered (see Figure 3b,d,f).

(3) If a critical point is added to the generating set, the resulting LDT skeleton is altered only to the extent that it acquires an additional branch; the previous skeleton geometry is unaltered (see Figure 3b,e,f).

(4) The LDT skeleton (for a simply connected object) is continuous.

Although comparison of the LDT and continuous skeletons is of theoretical interest, the LDT is an independent development with distinct image analysis potential. The fact that it can also serve as an approximation to the continuous skeleton is convenient, but not an essential property.

VI DETECTING THE CRITICAL POINTS OF AN OBJECT BOUNDARY

In the preceding section, we described the role played by the semantically meaningful critical points on the boundary of an object in generating the skeletal abstraction. Thus, the question of how such points can be automatically detected is of some interest. Because the issue of semantics goes beyond the visible content of the image, we must obviously limit our discussion here to finding critical points that can serve as feasible candidates for the desired shape landmarks, the final selection among these candidates being dependent on the intended use of the resulting skeleton.

A significant body of literature already exists for finding critical points based on a direct examination of the local curvature along the boundary, and indeed, for the direct characterization of shape based on partitioning the boundary into segments at these points (see Attneave and Arnoult [1966], and Freeman [1978]). The basic concept here is to approximate the immediately adjacent boundary on both sides of the point being considered by straight line segments, and if the angle between these line segments deviates sufficiently from 180 degrees, the given point is identified as a critical point. Numerous problems prevent this basic concept from performing in a satisfactory manner, and thus it must be augmented in various ways (see Davis [1977], Freeman [1978], Philbrick [1968], and Rosenfeld and Weszka [1975]).

One variation of the boundary analysis approach to finding critical points, which seems effective for locating sharp elbows and the endpoints of line segments, is to measure the distance between pairs of points on the boundary separated by fixed intervals of arc-length (these intervals must be matched to the maximum expected width of the lines being processed). At any placement on the boundary at which the distance is less than some fixed threshold, a critical point (endpoint or elbow) has been detected at the center point of the interval. Figure 4a shows some examples of the application of this technique to line type figures. In the particular implementation illustrated here, line widths of from one to ten pixels can be handled, and the thresholds are set so that "wedges" opening at angles of greater than 45 degrees are rejected; contiguous points on the boundary that satisfy the threshold criteria are assumed to derive from a single critical point at the center of their span.

It is interesting to observe that the distance-transform/skeleton machinery can provide an approach to the detection of critical points distinct from the boundary analysis techniques. For example, the end-segments of a skeleton generated in the conventional way (i.e., without using a labeled boundary) "point" to critical points on the boundary (see Figure 3a). Another method for locating critical points is to use an inverse form of the VDT. That is, for each point P on the boundary, we count the number of points external (or internal) to the boundary that are closer to P than to any other boundary point. Those boundary points that have large counts are the critical points. This approach, illustrated in Figures 4b,c,d, was made more effective by weighting the counting process by the inverse of the distance to the contributing point; because the VDT both associates an external point with a unique boundary point, and also provides the Euclidean distance, the weighted counting procedure is a computationally inexpensive addition.

There are two problems with direct application of distance transform methods for locating critical points. First, we note that on some sections of boundary having gently varying curvature (e.g., the

numerals 0 and 6 in Figure 4d), we may either get too many or too few critical points, depending on threshold setting. For some applications, it might be necessary to detect such situations, and iteratively adjust the local threshold until the desired number of critical points were located (e.g., we might require exactly one critical point in a region over which there is a sign change in curvature). Next, problems arise from "self masking" and from interference with nearby contours when the image is quantized at low resolution. Figures 4b and 4c contain relatively large isolated objects, and the obvious critical points are readily found; Figure 4d, with considerably lower effective resolution (and thus interference from nearby objects), does not produce as clean a result when processed with the same parameters. The same distance weighting function and critical point threshold was used in all the examples of Figure 4; in a crude sense these parameters were adjusted so that a critical point is a boundary point that "captures" at least a 45 degree wedge of local space.

An area for future investigation, which seems to have considerable potential, is the selective use of the inverse VDT (described above) as an alternative to convolution for image smoothing. Both boundary analysis and convolution techniques are "size" dependent: if the object size can vary between wide limits, the number of required "masks" and corresponding computation increase accordingly. The inverse VDT technique is essentially size independent, aside from the very low resolution problem noted above. Having once computed the transform, a much cheaper process than convolution with even a single mask, the portion of the image that we want to influence the computation of a point attribute can be simply adjusted by changing the distance weighting function; the transform does not have to be recomputed. A second disadvantage of convolution avoided by the inverse VDT is the problem of "blurring" due to smoothing across object boundaries. The partitioning property of the VDT (each point in the image is associated with exactly one of a set of distinguished/labeled points) can be used to ensure that natural boundaries are observed; e.g., by using the barrier technique elaborated on in the following section.

VII SKETCH COMPLETION

An important problem in scene analysis is that in analyzing an image, it is often the case that we can reliably detect some of the the edges, portions of homogenous regions, and pieces of coherent objects, while the remainder of the image content is ambiguous. For many purposes a completed representation of the scene is necessary, even if the completion process requires estimation of the missing pieces; e.g., to obtain a set of boundaries and regions that is geometrically similar in appearance to the edges and objects in the scene, or to establish adjacency in deriving a symbolic description of the scene. This completion process is called smoothing or sketching.

For the process of sketch completion, we assign a common label to every pixel identified as belonging to the same coherent region or object. We assign a special label to denote the presence of an edge. When the LDT encounters a pixel with an edge label, it neither changes this label nor allows it to propagate to adjacent pixels; thus, the labeled regions on both sides of the edge have the opportunity to grow out to the edge even though they were not equally displaced from it in the input image.*

Figure 5b shows the initial analysis of the 2-D projection of a scene (Figure 5a) for which we had both range and intensity data. We have developed analysis techniques for partitioning range/intensity scenes (to be described in a paper now in preparation) that produce excellent results over those portions of a scene where they are able to operate, but they will generally leave parts of the scene unlabeled. One motivating factor leading to the development of the LDT was to

---

* Because the edge feature blocks the normal propagation machinery, the LDT using the city-block distance metric may not converge after two passes (four passes are the minimum required when using the Euclidean distance metric VDT). Although this would generally not be case for typical scenes, additional passes may be necessary to ensure correct labeling of all points when edge-labeled features are present; convergence is reached when no distance (or label) changes occur in a pass through the algorithm.

develop a technique that could complete the labeling process in a reasonable way without excessive computation. Given the partial scene labeling shown by the colors in Figure 5b, we were able to apply the LDT to obtain the sketch shown in Figures 5c and 5d.

The only difference between the methods used to obtain an internal skeleton of a single object (as described in the first section of this paper), and an external skeleton or sketch, is the way in which the image is labeled; after the labeling process has been completed, and regardless of the application, the unaltered LDT algorithm is applied to the image to obtain the desired result.

We will now illustrate, with simple examples, two main mechanisms contributing to the result shown in Figure 5: region growing and boundary sketching.

The basic region growing operation of the LDT is shown in Figure 6a; a region "coloring" algorithm of the type described in Firschein and Fischler [1978] was first used to assign a common label to all of the pixels of each individual numeral, and distinct labels to the distinct numerals. The LDT then symmetrically partitioned the space about the labeled pixels (no edge labels were used), and the skeleton defining the partition was obtained by tracing the boundaries between the grown regions. One application of this simple mode of operation of the LDT is to establish nearest neighbor "adjacency" between spatially distributed objects that are not in contact with each other.

A constrained form of region growing is illustrated in Figure 6b. Explicit barrier or edge labels (which typically would correspond to previously detected object edges) are used to block the propagation mechanism of the LDT. Figure 6b shows an "S" shaped channel whose sides were assigned edge labels (colored white). Two differently labeled points (one "colored" gray, the other black) were placed at the opposite openings of the channel. The city block metric LDT was applied to the image, and the skeletal points, tracing the boundary between the grown gray and black regions, were also colored white. The way in which region growing was moderated by the blocking effect of the original edge-labeled points is apparent.

A boundary sketching mechanism of the LDT, illustrated in Figures 6c and 6d, achieves its result by extrapolating from differently labeled laterally adjacent line segments which define segments of a boundary (region growing using a similar labeling scheme for laterally adjacent line segments, but a different technique for propagation, is described in Strong and Rosenfeld [1973]). Figure 6c shows portions of a boundary between a white "T" shaped object and a gray background. Although the hard to see segment pair at the top center of the T is functionally adequate, the other segment pairs were made thicker to enhance visibility. The Euclidean metric LDT applied to Figure 6c produced the result shown in Figure 6d.

Obtaining smooth, intuitively correct boundary extrapolation is considerably more complicated than Figures 6c and 6d would tend to imply. First, assuming no interference from nearby objects, it is primarily growth from endpoints of the line segments that determines the shape of the extrapolated boundary. Second, given two differently labeled points (e.g., the adjacent endpoints of the laterally adjacent line segments), they will grow a straight line boundary corresponding to their perpendicular bisector. These two observations lead to the following conclusions:

(1) If we wish to get an intuitively acceptable extrapolation of a segment of a boundary (or interpolation between two or more widely separated segments), then the orientations of the line segment endpoint pairs must either be selected or adjusted to be perpendicular to the general orientation of the associated segment. If we use "thick" line segments (as was the case in Figure 6c), then these line segment ends must be carved into appropriate shapes.

(2) The uniform propagation mechanism of the LDT can only produce a straight line extrapolation.

On a continuous space, it is always possible to orient two points, arbitrarily close together, so that their perpendicular bisector has any desired orientation. On a uniform digital grid, 8-adjacent points can only produce perpendicular bisectors with orientations at integral multiples of 45 degrees. To get arbitrary orientations, the generating points must be separated; more acceptably, the "generator" could be an

angular "notch" with differently labeled sides. If the notch is made large enough, its angle bisector can be made to assume any desired direction.

Figure 7a shows the partial boundary segment pairs defining a triangle in which the ends of the segments were left to assume some ragged shape. The unacceptable result, after applying the Euclidean metric LDT, is shown in Figure 7b. When the line segment ends were carved into properly oriented notches, as shown in Figure 7c, the intuitively desirable result shown in Figure 7d was produced.

It should be noted that none of the endpoint adjustments discussed above were employed to obtain the results shown in Figure 5; there were enough edge-labeled points and partially "colored" regions, so that any interpolation due to boundary gaps did not get out of hand.

As an area for future investigation, it is interesting to observe that curved line extrapolation can be accomplished by a simple addition to the LDT. If we add, as an additional component to the label attached to each pixel, a number (or even a function) that can be used to control the speed of propagation of the growth process, then the differential growth of the laterally adjacent line segment pairs (and especially their "transmitting" ends) can be set to obtain any desired curvature in the extrapolated boundary.

VIII CONCLUDING COMMENTS

In this paper, we have shown how labeled information (semantic markers), appended to an image, can be used to control the smoothing/sketching behavior of distance-transform-generated skeletons and region boundaries. Automatic mechanisms for introducing semantically motivated constraints on the topology of the resulting skeleton or sketch were described; a key point is that it is not necessary to provide the initial semantic labeling at every location in the image, but only at those few critical location where there are

significant changes or discontinuities. The authors see this work both as a further clarification of the relationship between the contour of an object and descriptors of its shape, and as a step in the direction of extending the role played by iconic techniques in the overall scene-analysis paradigm (see Fischler [1978]).

Two promising areas for further development of the techniques discussed in this paper are: (1) associating "rate-of-propagation" control information with each of the labeled regions (as indicated briefly in the section on sketch completion)--this type of information would permit both curved interpolation of partial boundaries, and could also reflect the 3-D orientation of the depicted surfaces when "growing" incomplete regions; (2) investigating the use of the inverse VDT as an alternative to convolution for selected image smoothing applications (as discussed in the section on finding critical points)--this technique offers an escape from some of the size dependency problems associated with using a predefined "mask", and correspondingly reduces the computational requirements.

Although most of the discussion in the body of this paper described the LDT in the context of the 2-D city-block metric distance transform, and showed some examples of its use with a 2-D Euclidean metric distance transform, the LDT can obviously be appended to any distance transform, or even computed directly as noted in Appendix B. We are currently looking at the possibility of applying the LDT directly to 3-D scene representations.

Figure 1: Steps in producing a skeleton using the Labeled Distance Transform (LDT).

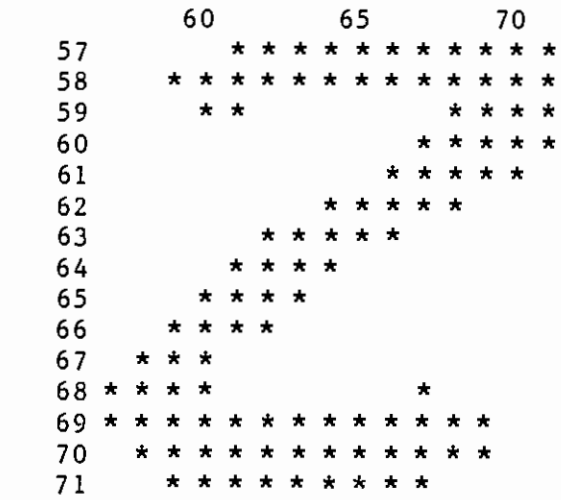


Figure 1a: Binary mask of input figure

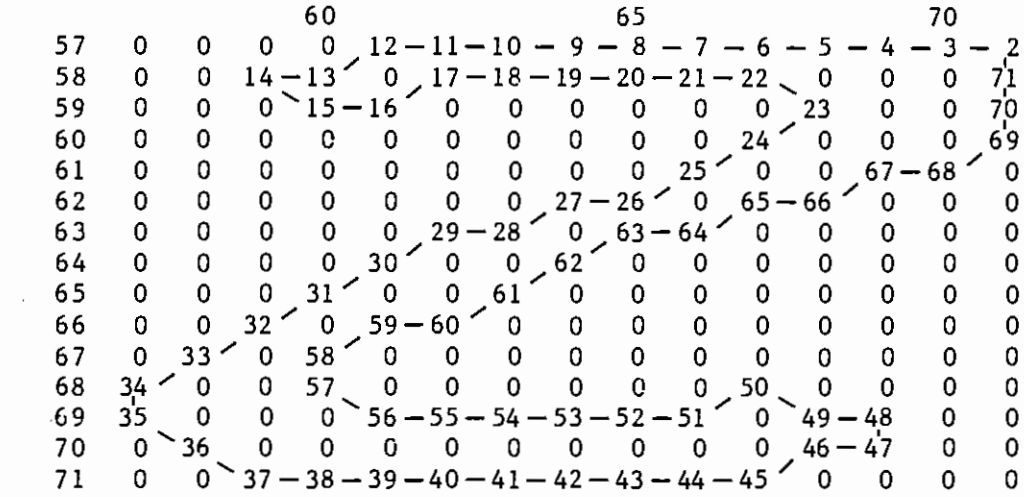


Figure 1b: Boundary points are sequentially labeled (using an algorithm similar to that given by Pavlidis [14]).

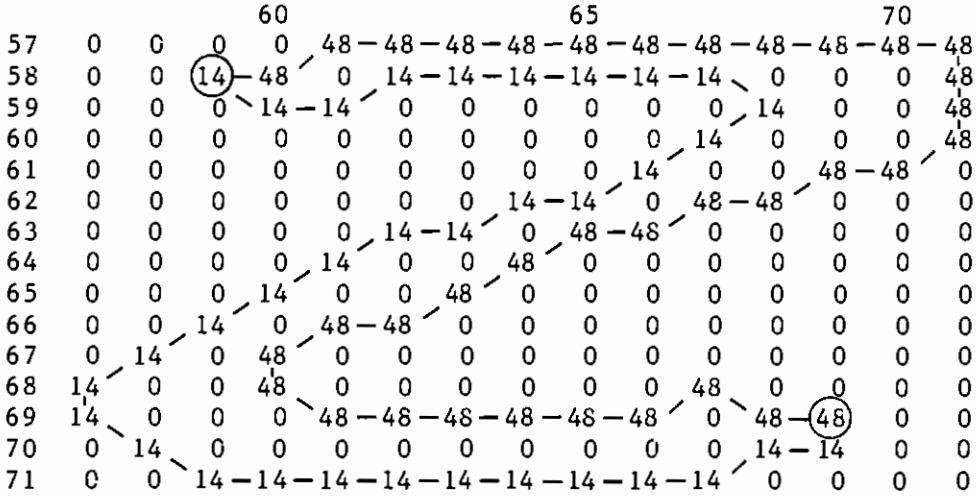


Figure 1c: Boundary points are relabeled based on the selection of critical points 14 and 48.

	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
55	7	6	5	4	3	3	2	2	2	2	2	2	2	2	2	2	2	3
56	6	5	4	3	2	2	1	1	1	1	1	1	1	1	1	1	1	2
57	5	4	3	2	1	1	0	0	0	0	0	0	0	0	0	0	0	1
58	4	3	2	1	0	0	1	0	0	0	0	0	0	1	1	1	0	1
59	5	4	3	2	1	0	0	1	1	1	1	1	1	0	1	1	0	1
60	6	5	4	3	2	1	1	2	2	2	2	1	0	1	1	1	0	1
61	7	6	5	4	3	2	2	2	2	1	1	0	1	1	0	0	1	2
62	8	7	6	5	4	3	2	1	1	0	0	1	0	0	1	1	2	3
63	7	6	5	4	3	2	1	0	0	1	0	0	1	1	2	2	3	4
64	6	5	4	3	2	1	0	1	1	0	1	1	2	2	3	3	4	5
65	5	4	3	2	1	0	1	1	0	1	2	2	3	3	4	4	5	6
66	4	3	2	1	0	1	0	0	1	2	3	3	2	3	3	4	5	6
67	3	2	1	0	1	0	1	1	2	2	2	2	1	2	2	3	4	5
68	2	1	0	1	1	0	1	1	1	1	1	1	0	1	1	2	3	4
69	2	1	0	1	2	1	0	0	0	0	0	0	1	0	0	1	2	3
70	3	2	1	0	1	1	1	1	1	1	1	1	1	0	0	1	2	3
71	4	3	2	1	0	0	0	0	0	0	0	0	0	1	1	2	3	4
72	5	4	3	2	1	1	1	1	1	1	1	1	1	2	2	3	4	5
73	6	5	4	3	2	2	2	2	2	2	2	2	2	3	3	4	5	6

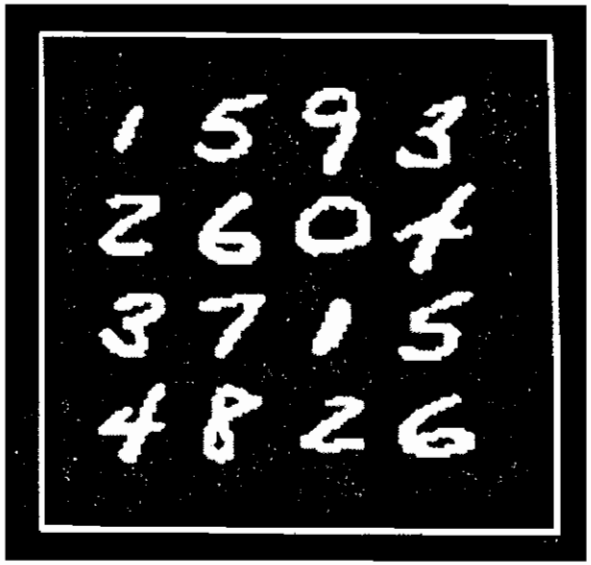
Figure 1d: Distance transform of Figure 1a (via city-block metric LDT).

```
    55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
55  14  14  14  14  14  48  48  48  48  48  48  48  48  48  48  48  48  48
56  14  14  14  14  14  48  48  48  48  48  48  48  48  48  48  48  48  48
57  14  14  14  14  14  48  48  48  48  48  48  48  48  48  48  48  48  48
58  14  14  14  14  14  48  48  14  14  14  14  14  14  48  48  48  48  48
59  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  48  48  48
60  14  14  14  14  14  14  14  14  14  14  14  14  14  14  48  48  48  48
61  14  14  14  14  14  14  14  14  14  14  14  14  14  48  48  48  48  48
62  14  14  14  14  14  14  14  14  14  14  14  14  48  48  48  48  48  48
63  14  14  14  14  14  14  14  14  14  14  48  48  48  48  48  48  48  48
64  14  14  14  14  14  14  14  14  14  48  48  48  48  48  48  48  48  48
65  14  14  14  14  14  14  14  48  48  48  48  48  48  48  48  48  48  48
66  14  14  14  14  14  14  48  48  48  48  48  48  48  48  48  48  48  48
67  14  14  14  14  14  48  48  48  48  48  48  48  48  48  48  48  48  48
68  14  14  14  14  48  48  48  48  48  48  48  48  48  48  48  48  48  48
69  14  14  14  14  14  48  48  48  48  48  48  48  48  48  48  48  48  48
70  14  14  14  14  14  14  48  48  48  48  48  48  14  14  14  14  14  14
71  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14
72  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14
73  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14
```

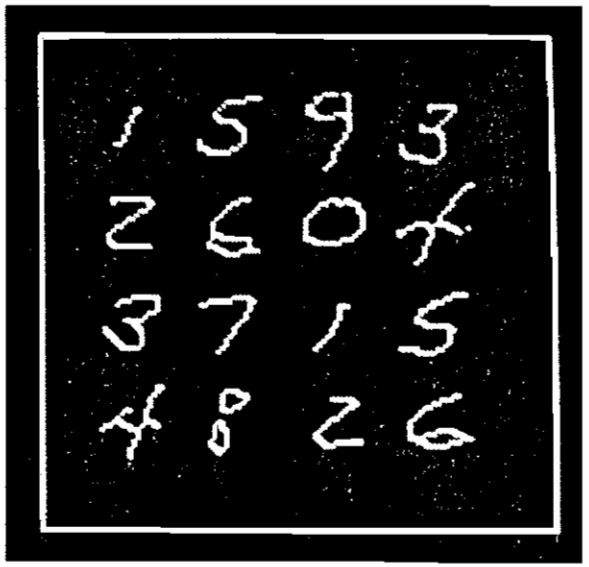
Figure 1e: Label transform of Figure 1a (via city-block metric LDT).

```
        60          65          70
57        * + + + + + + * * * *
58    X + + * * * * * * + + + * *
59      * *                 * + * *
60                        * + * * *
61                      * + * * *
62                  * + + * *
63              * * + * *
64            * + + *
65          * + * *
66        * + * *
67    * + *
68 * + + *                 *
69 * * + + * * * * * * * + + X
70   * * + + + + + + + * * *
71     * * * * * * * * *
```

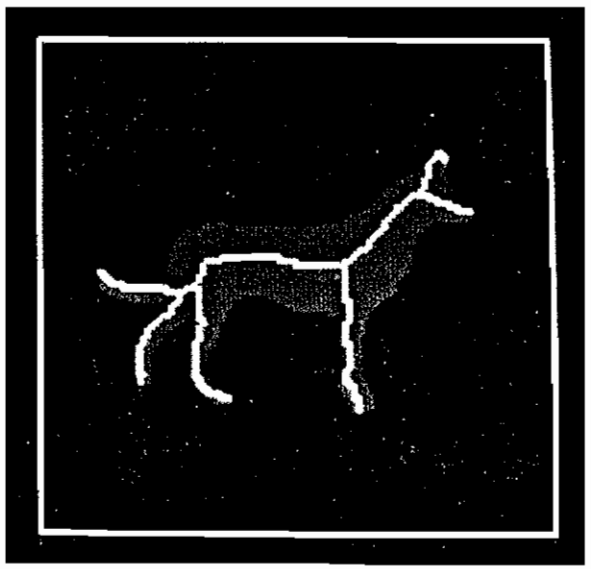
Figure 1f: Composite image showing the original object mask (*), the two critical points (X), and the resulting city-block metric LDT skeleton (+).



(a) ORIGINAL IMAGE CONTAINING HANDWRITTEN NUMERALS.

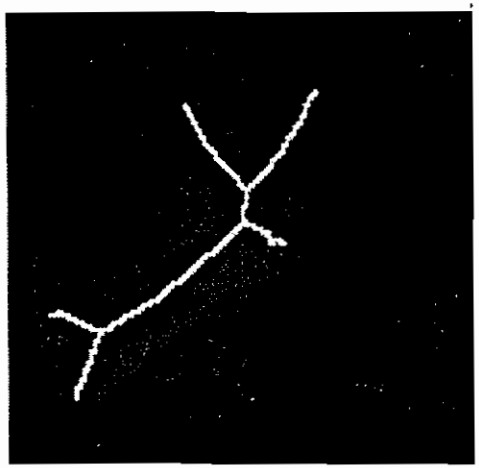


(b) SKELETON GENERATED FOR NUMERALS USING THE CITY-BLOCK METRIC LDT.

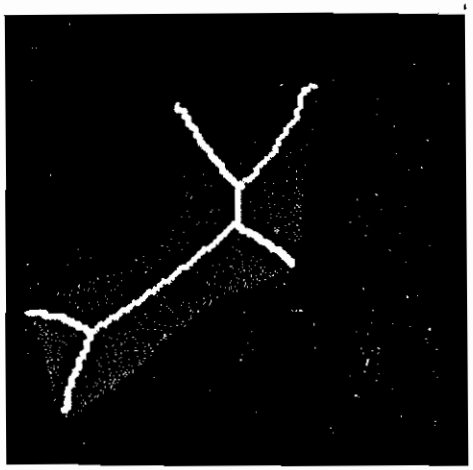


(c) IMAGE OF A DOG (traced from Philbrick [1968]) AND ITS SKELETON GENERATED BY THE EUCLIDEAN METRIC LDT.

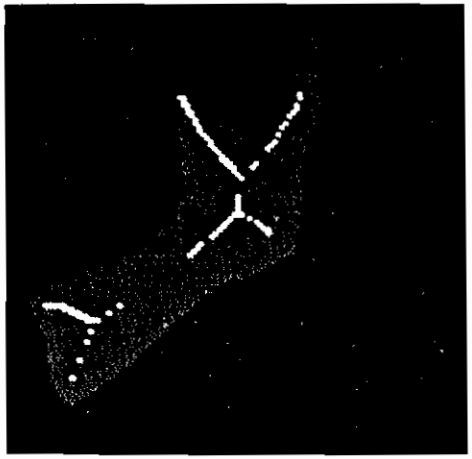
FIGURE 2 EXAMPLES OF LDT PRODUCED SKELETONS



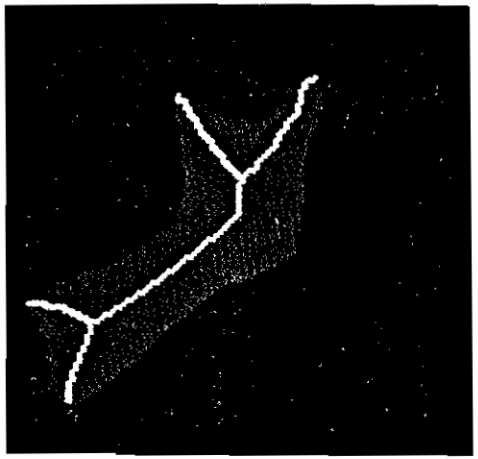
(a) SILHOUETTE OF HUMAN JAWBONE (mandible) AND ITS CONTINUDUS SKELETON — TRACED FROM BOOKSTEIN [1979].



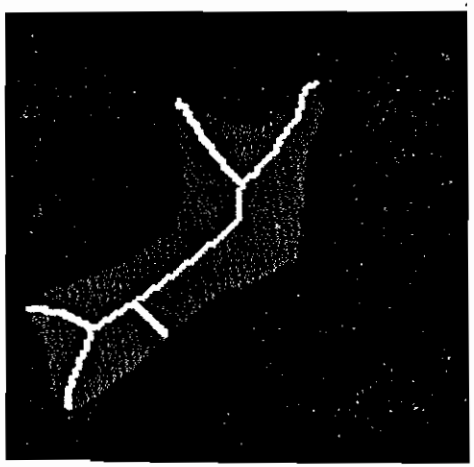
(b) MANDIBLE WITH LDT GENERATED SKELETON.



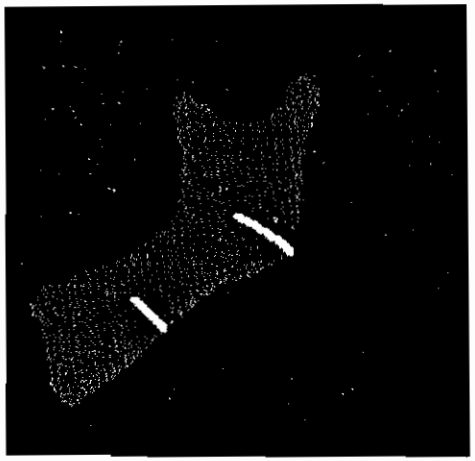
(c) MANDIBLE SHOWING COMMON POINTS OF THE CONTINUOUS AND LDT SKELETONS; THE REMAINING SKELETAL POINTS DIFFERED BY APPROXIMATELY ONE PIXEL DUE TO TRACING AND QUANTIZATION ERRORS.



(d) MANDIBLE WITH LDT SKELETON GENERATED AFTER REMOVING ONE CRITICAL POINT.

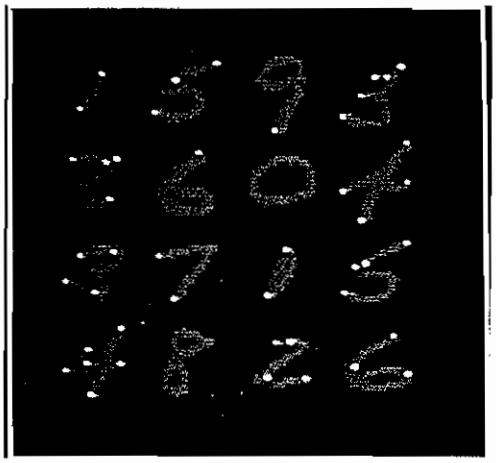


(e) MANDIBLE WITH LDT SKELETON GENERATED AFTER SHIFTING ONE CRITICAL POINT.

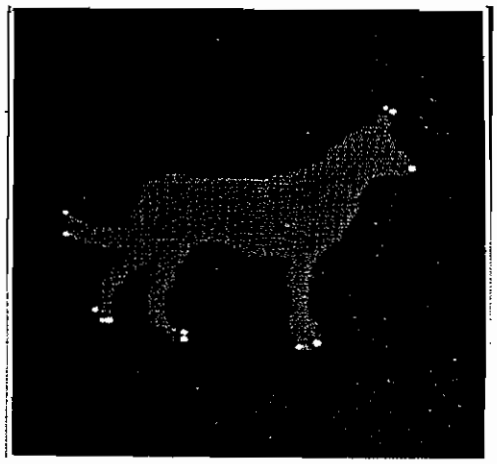


(f) MANDIBLE SHOWING DIFFERENCES BETWEEN SKELETONS IN FIGURES (3b) AND (3e). NOTE THAT ONLY THE BRANCH ASSOCIATED CRITICAL POINT IS SHIFTED; THE REMAINING PORTION OF THE SKELETON IS LEFT UNCHANGED.

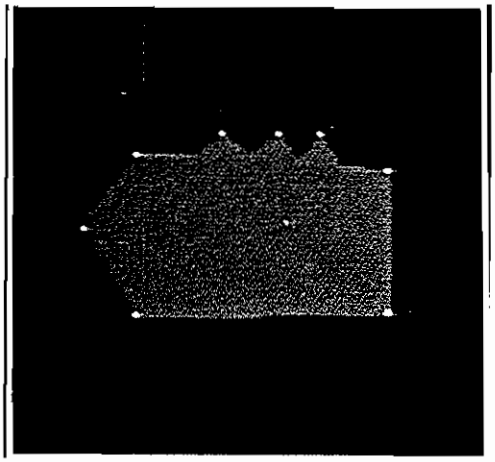
FIGURE 3 COMPARISON OF CONTINUOUS AND EUCLIDEAN METRIC LDT GENERATED SKELETONS



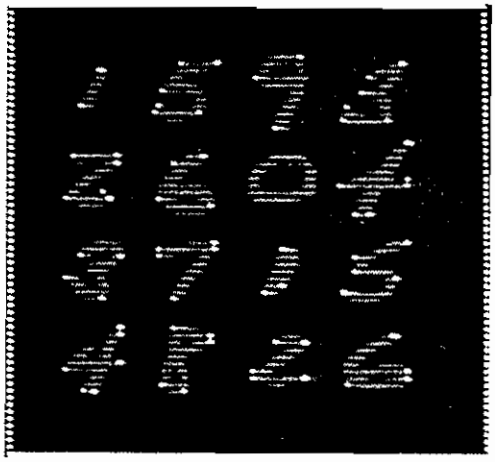
(a) CRITICAL POINTS LOCATED BY A BOUNDARY ANALYSIS TECHNIQUE WHICH FINDS SHARP ELBOWS AND ENDPOINTS OF LINE SEGMENTS.



(b) CRITICAL POINTS LOCATED BY A WEIGHTED "INVERSE VDT" TECHNIQUE.

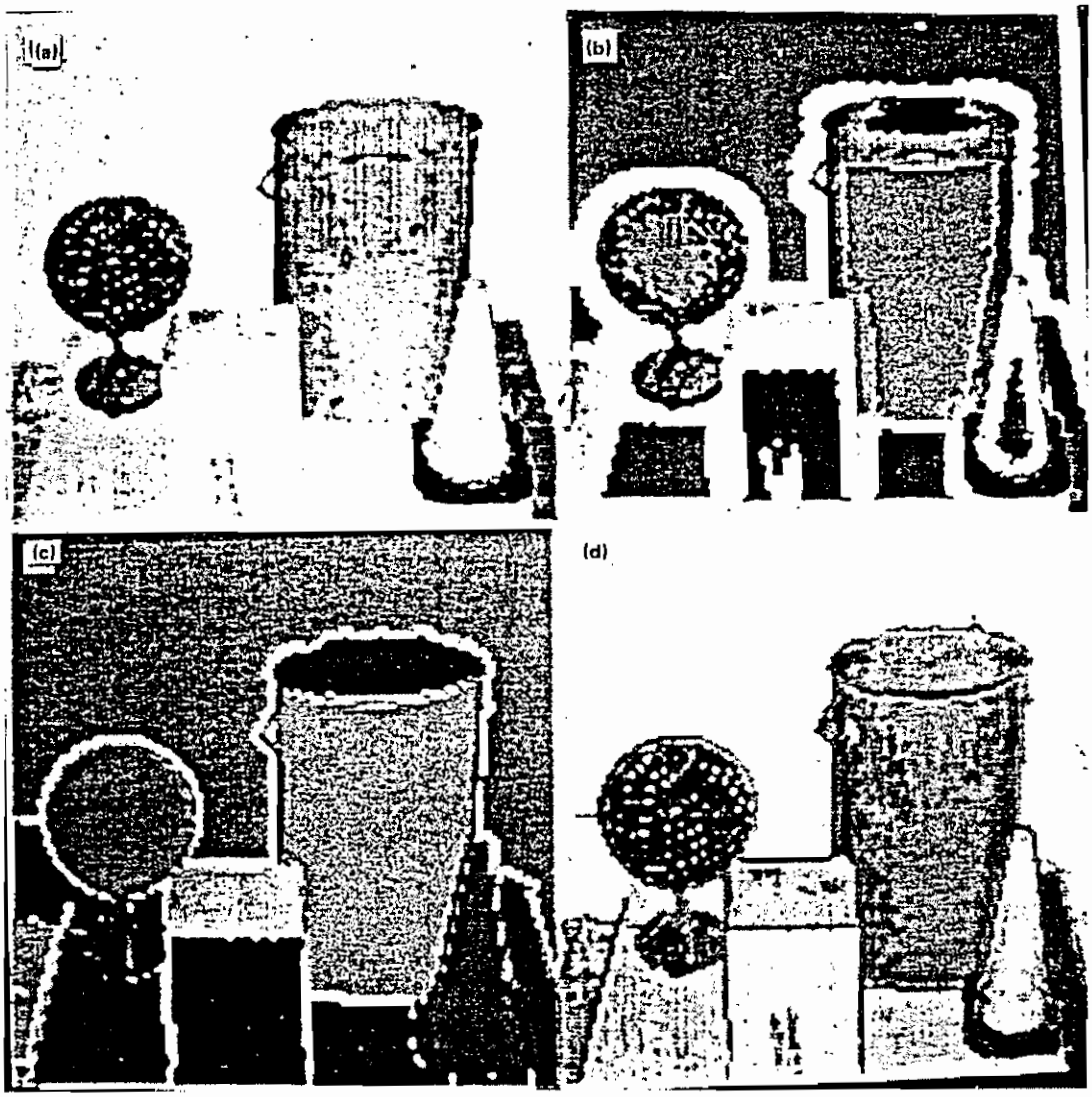


(c) CRITICAL POINTS LOCATED BY A WEIGHTED "INVERSE VDT" TECHNIQUE.



(d) CRITICAL POINTS LOCATED BY A WEIGHTED "INVERSE VDT" TECHNIQUE.

FIGURE 4 LOCATING CRITICAL POINTS ON OBJECT BOUNDARIES



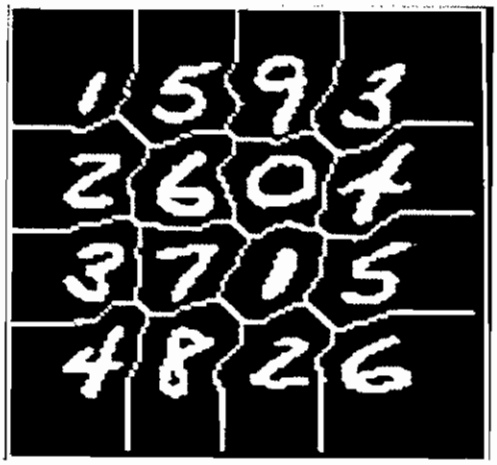
(a) THE IMAGE OF A 3-DIMENSIDNAL SCENE FOR WHICH WE HAVE BOTH RANGE AND INTENSITY DATA.

(b) AN INITIAL ANALYSIS OF THE SCENE PARTITIONED INTO SURFACES AND OBJECTS AS INDICATED BY THE DIFFERENT COLORS.

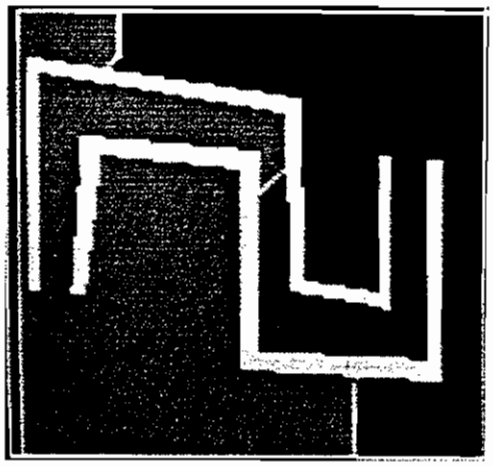
(c) EXPANDED REGIONS OBTAINED BY USING THE LDT WITH THE BARRIER (edge) FEATURE.

(d) EDGES SKETCHED BY THE LDT.

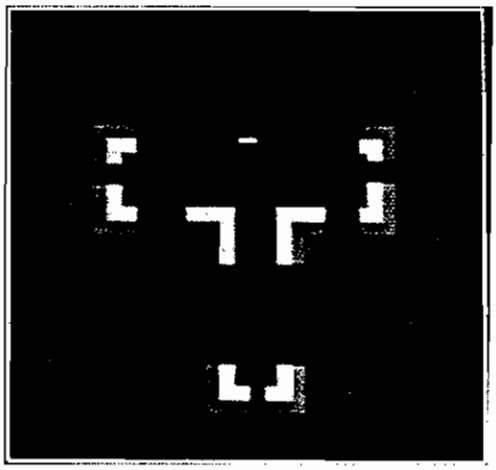
FIGURE 5 USE OF THE LDT FOR SKETCH COMPLETION



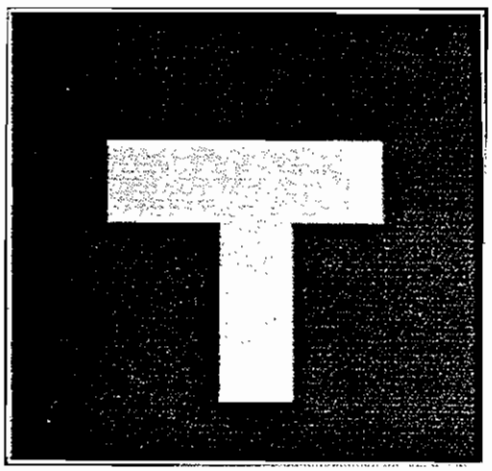
(a) EUCLIDEAN METRIC LDT PARTITION OF AN IMAGE BY "GROWING" THE REGIONS CORRESPONDING TO THE DISTINCTLY LABELED NUMERALS.



(b) CONSTRAINED REGION GROWING THROUGH THE USE OF EDGE (barrier) LABELED POINTS.

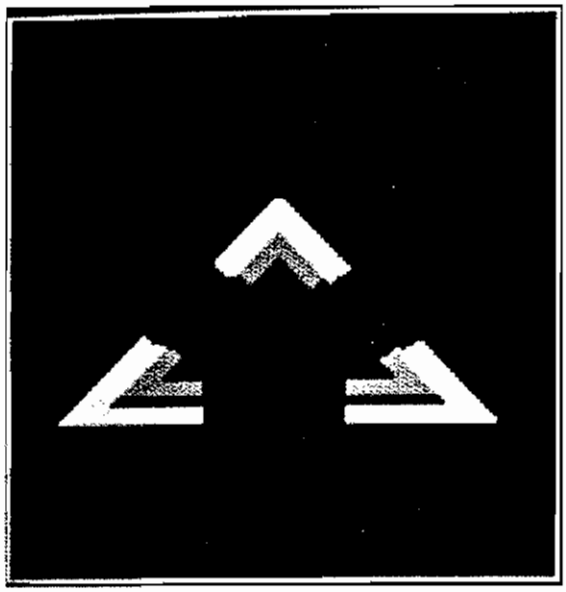


(c) SEGMENTS OF THE BOUNDARY OF A "T" SHAPED OBJECT DEFINED BY DIFFERENTIALLY LABELED LATERALLY ADJACENT LINE SEGMENTS.

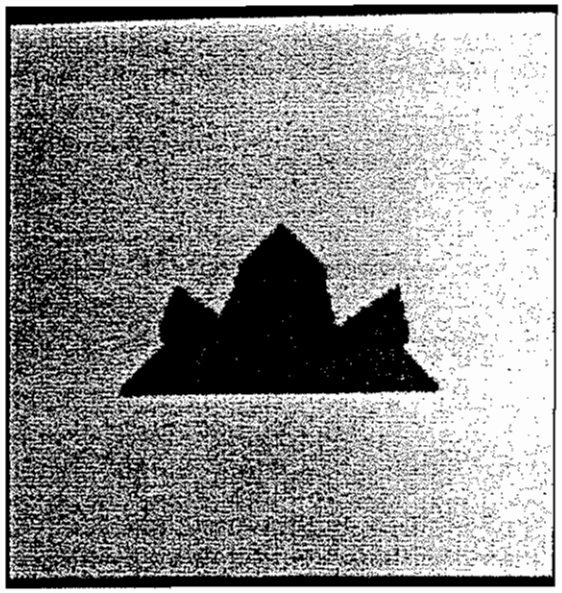


(d) RESULT OF APPLYING THE EUCLIDEAN METRIC LDT TO FIGURE (6c).

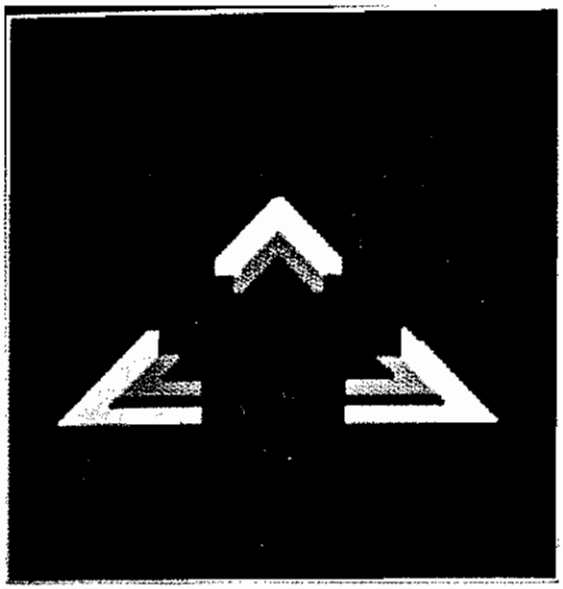
FIGURE 6 ILLUSTRATION OF THE LDT MECHANISMS FOR REGION GROWING AND BOUNDARY SKETCHING



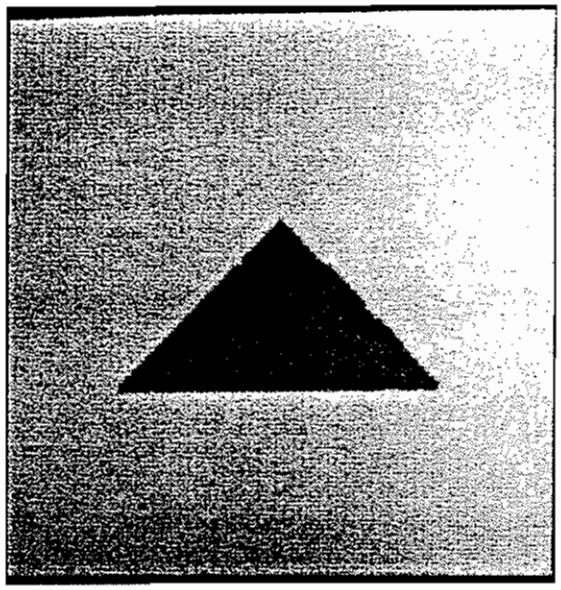
(a) DIFFERENTIALLY LABELED SEGMENTS OF THE BOUNDARY OF A TRIANGLE. SEGMENTS HAVE RAGGED ENDS.



(b) RESULT OF APPLYING THE EUCLIDEAN METRIC LDT TO FIGURE (7a).



(c) NOTCHES CUT IN ENDS OF DIFFERENTIALLY LABELED BOUNDARY SEGMENTS TO OBTAIN SMOOTH EXTRAPOLATION.



(d) RESULT OF APPLYING THE EUDLICEAN METRIC LDT TO FIGURE (7c).

FIGURE 7 THE USE OF "TRANSMITTERS" TO OBTAIN SMOOTH BOUNDARY COMPLETION VIA THE LDT

REFERENCES

1. F. Attneave and M. D. Arnoult, "The Quantitative Study of Shape and Pattern Perception," Pattern Recognition, L. Uhr, ed., pp. 123-141 (Wiley, 1966).

2. H. Blum, "A Transformation for Extracting New Descriptors of Shape," Models for the Perception of Speech and Visual Form, Walthen Dunn, ed., pp. 362-380, MIT Press (proc. of meeting held Nov. 1964), (1967).

3. H. Blum and R. N. Nagel, "Shape Description Using Weighted Symmetric Axis Features," Pattern Recognition, Vol. 10, No. 3, pp. 167-180 (1978).

4. F. L. Bookstein, "The Line Skeleton," (report prepared for the Center for Human Growth and Development), University of Michigan, (March 1979).

5. F. L. Bookstein, "The Measurement of Biological Shape and Shape Change," (Springer-Verlag, 1978).

6. L. Calabi and W. E. Hartnett, "Shape Recognition, Prairie Fires, Convex Deficiencies and Skeletons," Am. Math. Mon., Vol. 75, p. 335 (1968).

7. L. S. Davis, "Understanding Shape: Angles and Sides," IEEE Trans. Comput., Vol. 26, No. 3, pp. 236-242 (1977).

8. O. Firschein and M. A. Fischler, "Association Algorithms for Digital Imagery," Conference Record of the Twelfth Annual Asilomar Conference on Circuits, Systems, and Computers, IEEE Catalog No. 78CH1369-8C/CAS/CS (November 1978).

9. M. A. Fischler, "On the Representation of Natural Scenes," in Computer Vision Systems, A. R. Hansen and E. M. Riseman, eds., pp. 47-52 (Academic Press, 1978).

10. H. Freeman, "Shape Description Via the Use of Critical Points," Pattern Recognition, Vol. 10, No. 3, pp. 159-166 (1978).

11. A. R. Hanson and E. M. Riseman, "Segmentation of Natural Scenes," in Computer Vision Systems, A. R. Hansen and E. M. Riseman, eds., pp. 129-163 (Academic Press, 1978).

12. U. Montanari, "A Method for Obtaining Skeletons Using a Quasi-Euclidean Distance," J. ACM, Vol. 15, No. 4, pp. 600-624 (October 1968).

13. U. Montanari, "Continuous Skeletons from Digitized Images," J. ACM, Vol. 16, No. 4, pp. 534-549 (October 1969).

14. T. Pavlidis, "A Minimal Storage Boundary Tracing Algorithm and its Application to Automatic Inspection," IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-8, No. 1, pp. 66-69 (January 1978).

15. J. L. Pfaltz and A. Rosenfeld, "Computer Representation of Planar Regions by their Skeletons," C. ACM, Vol. 10, No. 2, pp. 119-122, 125 (February 1967).

16. O. Philbrick, "Shape Description with the Medial Axis Transformation," in Pictorial Pattern Recognition, pp. 395-407 (Thompson Book Co., 1968).

17. L. Quam, "Euclidean Distance Transforms on Binary Images" (in preparation).

18. A. Rosenfeld, R. A. Hummel, and S. W. Zucker, "Scene Labeling by Relaxation Operations," IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-6, No. 6, pp. 420-433 (June 1976).

19. A. Rosenfeld and J. L. Pfaltz, "Sequential Operations in Digital Picture Processing," J. ACM, Vol. 13, No. 4, pp. 471-494 (October 1966).

20. A. Rosenfeld and J. S. Weszka, "An Improved Method of Angle Detection on Digital Curves," IEEE Trans. Comput., Vol. C-24, No. 9, pp. 940-941 (September 1975).

21. M. I. Shamos and D. Hoey, "Closest-Point Problems," Proc. 16th IEEE Symp. on Foundations of Computer Science, pp. 151-162 (1975).

22. J. P. Strong III and A. Rosenfeld, "A Region Coloring Technique for Scene Analysis," Comm. ACM, Vol. 16, No. 4, pp. 237-245 (April 1973).

23. J. M. Tenenbaum and H. G. Barrow, "IGS: A Paradigm for Integrating Image Segmentation and Interpretation," Pattern Recognition and Artificial Intelligence, C. H. Chen, ed., pp. 472-507 (Academic Press, 1976).

24. D. L. Waltz, "A Parallel Model for Low-Level Vision," in Computer Vision Systems, A. R. Hansen and E. M. Riseman, eds., pp. 175-186 (Academic Press, 1978).

Appendix A

The Basic Labeled Distance Transform (LDT) Algorithm for the City-Block Distance Metric

Assume two arrays: "label" and "dist;" label has non-zero values which are to be propagated into zero locations. A special barrier label (B) may also be present. The distance array could be initialized during the first forward pass, but for clarity we assume that dist has the following values:

```
 0 in label <=> dist has value #    (#  >> 0)
 B in label <=> dist has value ↑↑   (↑↑ >> #)
other label <=> dist has value 0
```

The algorithm converges after exactly one forward and one backward pass (one iteration) if barrier labels are not used. If barrier labels are used, additional iterations may be required.

The algorithm for generating the skeleton is presented separately for clarity. It could be executed as part of the backward pass on the final iteration.

FIGURE A-1: Forward Pass, LDT algorithm.
For each [i,j] from top to bottom and from left to right.

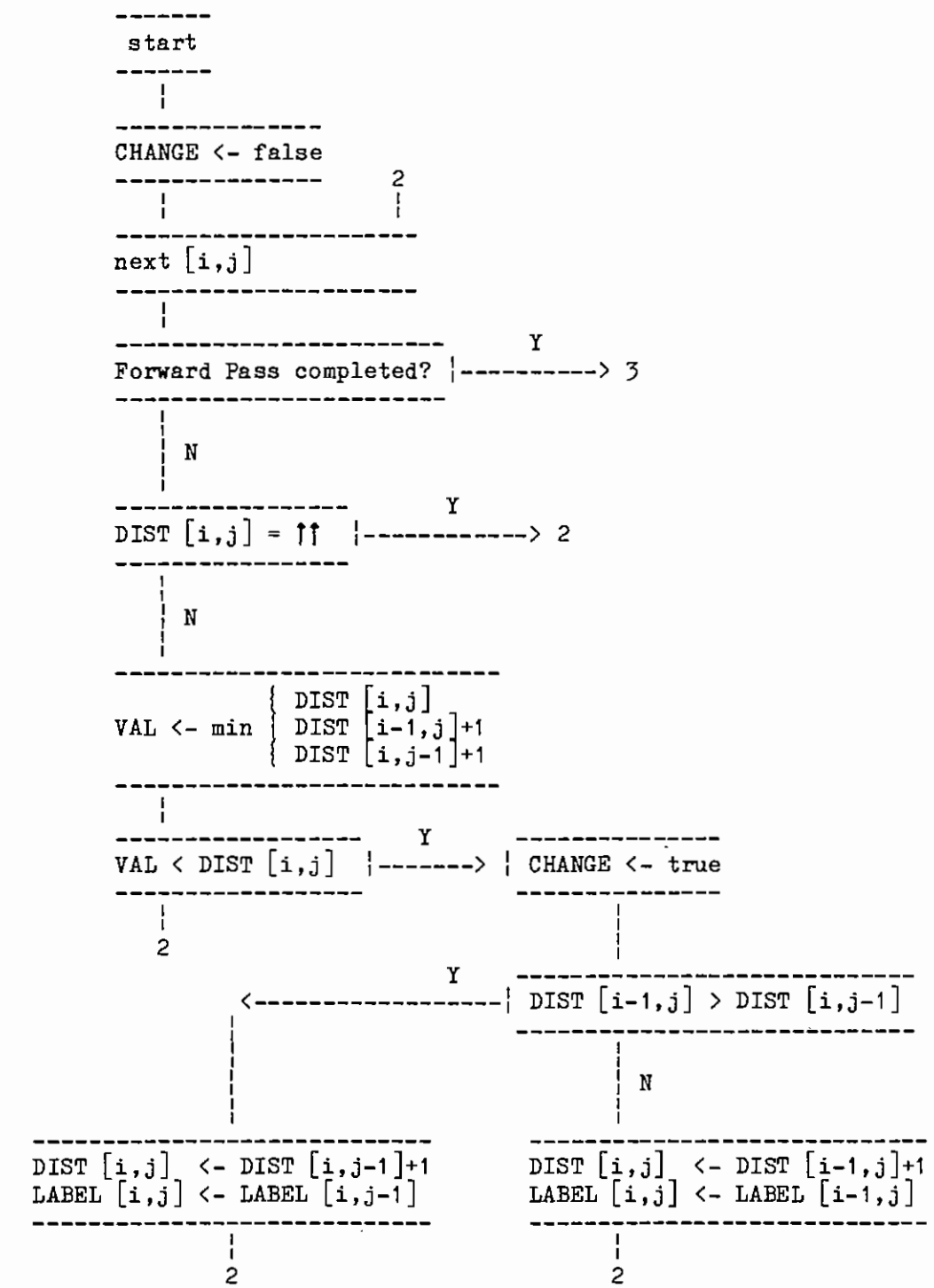


FIGURE A-2: Backward Pass, LDT Algorithm.
For each [i,j] from bottom to top and from right to left.

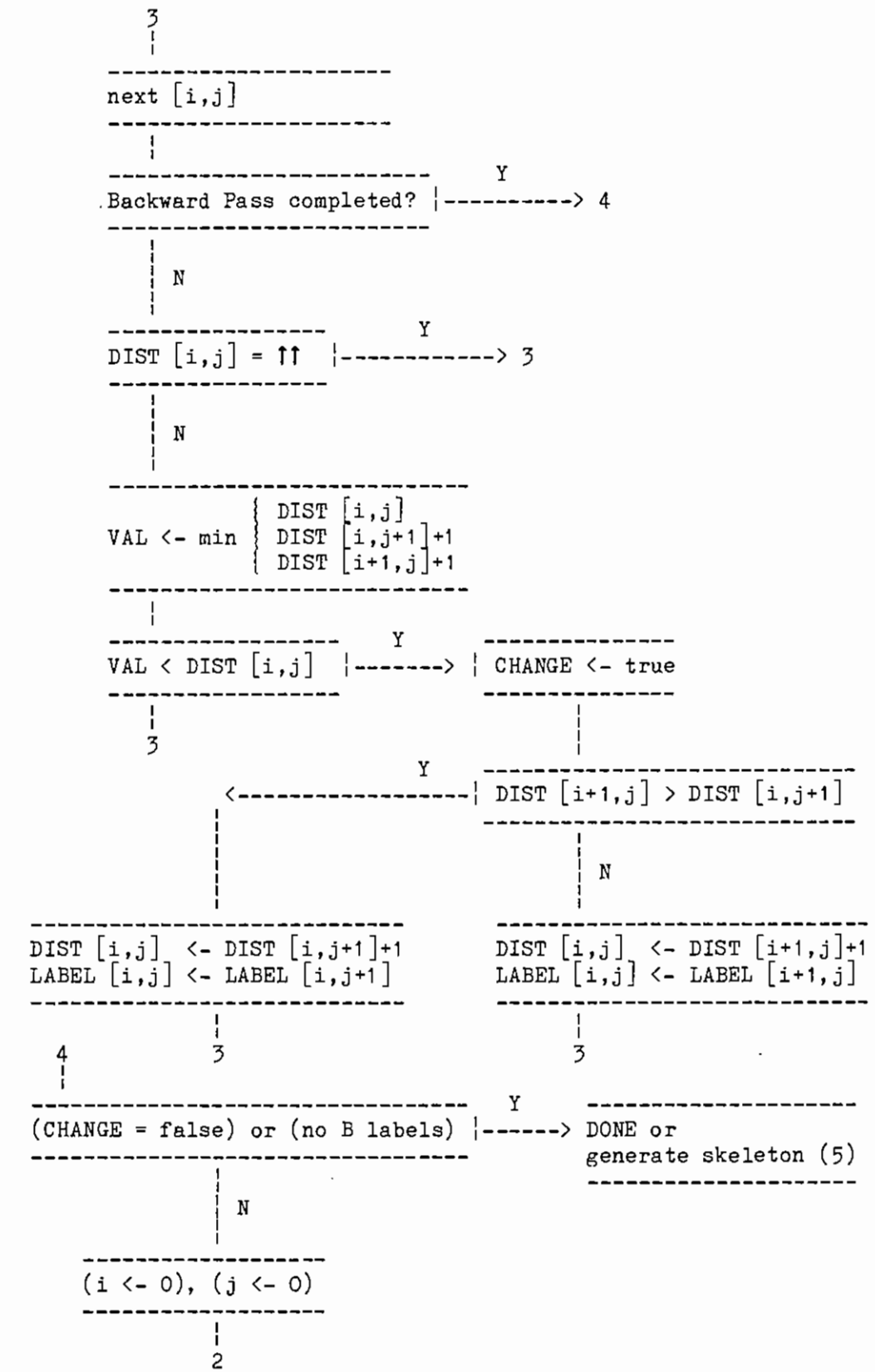
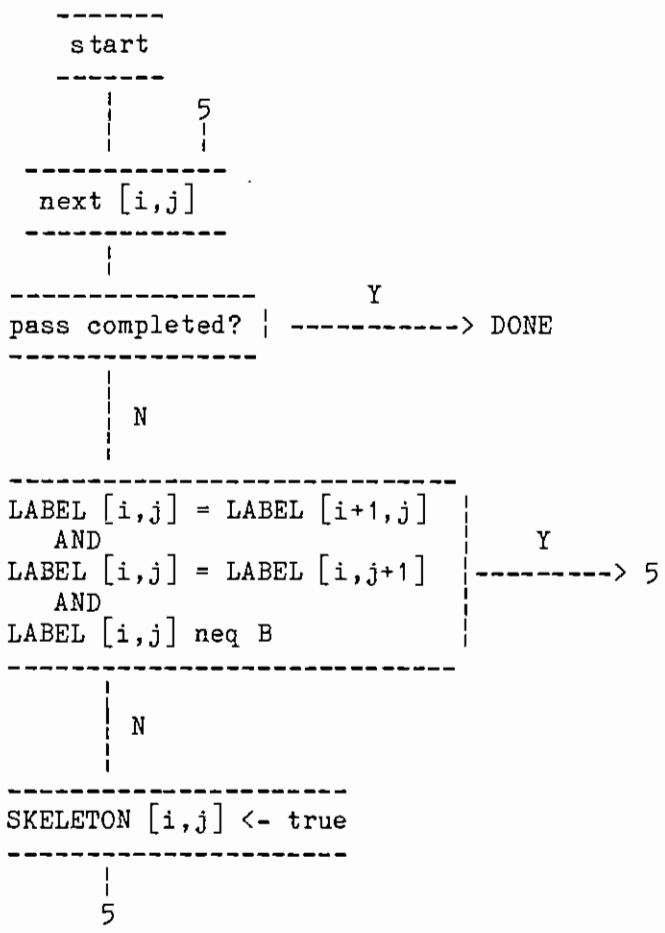


FIGURE A-3: Skeleton Generation, LDT Algorithm.
For each [i,j] from bottom to top and from right to left, using LABEL array as generated in final backward pass of the LDT algorithm.



Appendix B

Properties of the LDT Skeleton (LDTS)

Given any planar set of distinguished points, called V-points, they will define a partition of the 2-D Eucledian plane into convex regions if each such region, called a V-region, contains all points in the plane that are closer to the corresponding V-point than to any other V-point in the distinguished set.

Let us now assume that all V-points are assigned labels, and that some V-points, whose corresponding regions have common boundaries, also have common labels. The union of the boundaries, after the boundaries between regions containing common labels have been deleted, is defined to be the LDT skeleton (LDTS). The corresponding partition of the plane will be called the LDT-partition, and the region associated with a particular label will be called an LDT-region.

Assume:

(1) Continuous Euclidean 2-D plane.

(2) V-points exist only on the closed continuous boundary, C, of a simply connected region.

(3) Closed continuous segments of C, bounded by distinguished V-points, called critical points, are assigned unique labels. Critical points are assumed to be "double points" in that they correspond to the superposition of the end-points of the two distinctly labeled adjacent boundary arcs.

(4) We only consider that portion of the LDTS delimited by C; i.e., the restriction of the plane to the closed area bounded by C.

Under the above assumptions, we can establish the following results:

LEMMA 1: Every uniquely labeled boundary segment of C is enclosed in an LDT-region of the LDT-partition of the plane, and forms a segment of the boundary of the LDT-region in the restriction of the plane to the area bounded by C.

Proof: The LDT-regions are the unions of adjacent (convex) V-regions, each V-region containing its corresponding V-point. Thus, each LDT-region will contain all its

V-points, which are assumed to lie on a continuous segment of C. Since we are only considering that portion of the Euclidean plane bounded by C, each segment of C must necessarily be a boundary of its associated LDT-region. QED

LEMMA 2: The LDTS, restricted to the area bounded by C, is a tree with all vertices of order one lying on critical points on C.

Proof: Since the LDTS delimits the boundaries of the LDT-regions, and boundary lines must exhibit closure or extend indefinitely, vertices of order one can only exist on C (in essence, where the LDTS is "chopped off"). Every LDTS vertex point on C must be of order one and lie on a critical point: since the boundary between LDT-regions "generated" by (or originating from) adjacent arcs on C must pass through their common point (X) of closure on C, and since the skeleton terminates on C (by the restriction assumption), and since only two distinctly labeled segments can be contiguous on C, X must be a vertex point of the LDTS of order one.

The LDTS, in the restriction to C, must be a tree. If this were not the case, there would be a circuit of the LDTS internal to C; this in turn implies that the LDT-region enclosed by the circuit contains none of its V-points (which are all assumed to lie on C), which is in violation of Lemma 1. QED

LEMMA 3: Each point q on the LDTS, which is not a vertex point, is a nearest neighbor (nn) of exactly two distinctly labeled subsets of adjacent V-points on C, and lies on the common boundary of the corresponding LDT-regions. A skeletal arc has a vertex of order one if and only if its two corresponding boundary arcs are adjacent on C.

Proof: Every point that is the nearest neighbor (nn) of two V-points must lie on the common boundary of their associated V-regions. If the two given V-points have distinct labels, the common boundary of their V-regions will be an arc of the LDTS (by definition). If a point of the LDTS is a nn of three or more distinctly labeled boundary points, then it lies on the boundary of at least three LDT-regions, and thus at the intersection point of at least three boundary lines, and hence is a vertex point of the LDTS.

Two distinctly labeled adjacent boundary arcs (meeting at a critical point, by definition) are the boundary arcs corresponding to the skeletal branch forming the boundary

between their two LDT-regions (as shown above); by Lemma 2, this critical point is a vertex of the skeleton of order one. QED

Lemma 3 establishes the following theorem:

THEOREM 1:

a) Each arc of the LDTS is generated by portions of exactly two distinctly labeled segments of C.

b) If an arc Q of the LDTS has a vertex of order one (i.e., if it terminates on a critical point X on C), then its generating arcs are adjacent on C and are separated by X.

c) If Q is an interior arc of the LDTS (i.e., both vertices have order two or more), then its generating arcs are not adjacent on C, i.e., they are not separated by a single critical point.

LEMMA 4: Let q be a point on the LDTS, and p one of its nn on C lying on a (single) uniquely labeled segment P between critical points CP1 and CP2 (see Figure B1). As q moves from CP1 to CP2 along those arcs (Q) of the LDTS that bound the LDT-region containing P, p will also move from CP1 to CP2. This movement will be monotonic with respect to distinct pairs of nn (q1,p1) and (q2,p2) given that (q1 neq q2) and (p1 neq p2). The movement of p will not necessarily be continuous, nor will p necessarily pass through every point on P, since there can be points on P which are not nn of any point q on the LDTS.

Proof: To prove monotonicity, assume that this is not the case. The straight lines connecting the point pairs (q1,p1) and (q2,p2) will then cross as shown in Figure B1. Now consider the quadrilateral (q1,q2,p1,p2); one of its four interior angles must equal or exceed 90 degrees since all four angles sum to 360 degrees. The diagonal of the quadrilateral opposite this non-acute angle must be longer than either of the two sides of the quadrilateral forming the angle. Thus, contrary to our original assumption, either p1 is not a nn of q1, or p2 is not a nn of q2. The contradiction establishes the proof. QED

Referring to the situation presented in Lemma 4, let us call the nn on C of vertex Va (for every vertex Va of order two or more) "virtual critical point" VCPa; if there is more than one such point on P for a

given Va, these points will be contiguous by Lemma 4, and then VCPa will be a segment rather than a point. Now consider the effect on the LDTS if CP1 was repositioned on C so as to be closer to CP2, assuming no additional changes to either the geometry of C or the location of the other critical points.

LEMMA 5: If CP1 is replaced by CP1', where CP1' lies between CP1 and VCP1, then the only portion of the LDTS whose geometry is altered by the resulting relabeling of C is that portion of arc Q between CP1 and V1 which has as nn on C the points between CP1 and CP1' (see Figure B1).

Proof: If a critical point X is displaced by some small interval of distance along the contour C, then by Lemma 2, the arc of the skeleton terminating at X will be similarly displaced. All points along the original skeletal arc Q, that had nn lying in the displacement interval D, also had distinctly labeled corresponding points on the other side of X along C (by Theorem 1a); the movement of X caused these pairs of distinctly labeled points to assume identical labels and thus no longer generate skeletal points. If q' is the furthest point from CP1 along Q that has CP1' as one of its nn on C, then no point in the interval D can be a nn of any point on the skeleton beyond q' (by Lemma 4). Thus, the rest of the skeleton remains unaltered by the labeling changes that occurred over the interval D of C by the movement of critical point X. QED

LEMMA 6: If CP1 is replaced by CP1', where CP1' lies between VCP1 and CP2, then the entire arc between CP1 and V1 will be replaced by a new arc extending from CP1' on C to a new vertex point on the LDTS. However, none of the other vertex points on the LDTS will be moved, nor will the geometry of any of the other arcs of the LDTS be altered (see Figure B1).

Proof: By the same argument used in Lemma 5, if q1 is the most distant point from CP1 along Q which is a nn of CP1', then only the arc from CP1 to V1, and the portion of arc from V1 to q1, can be altered by the replacement of critical point CP1 by CP1'.

Lemma 5 establishes the fact that the arc from CP1 to V1 will be displaced. Since q1 is a nn of CP1', it will be the new vertex point for the arc of the skeleton leaving the contour C at critical point CP1'; this is because q1 must lie on the boundary between the two distinctly labeled regions corresponding to the adjacent boundary arcs terminating at CP1'.

The labeling change that was induced on C by the movement of CP1 to CP1' cannot result in the portion of C between

arc of C that contributed to the generation of the arc of the LDTS between V1 and V2 (Theorem 1c). Thus, each point of the LDTS between V1 and q1 will still have the same pair of distinctly labeled nn on C, and will remain unaltered. QED

LEMMA 7: If CP1 is deleted (equivalent to being moved until it coincides with CP2), then the arc between CP1 and V1 will be deleted. The geometry of the remainder of the LDTS will be left unaltered.

Proof: Removal of a critical point X causes the two adjacent boundary arcs terminating at X to receive common labels, and thus eliminates the common boundary between their LDT-regions which was an arc of the LDTS. By Theorem 1c, no other arc of the LDTS is affected by this labeling change. QED

We have thus established the following theorem:

THEOREM 2:

a) If a critical point is deleted, then its associated arc will also be deleted. The geometry of the remainder of the LDTS will remain unaltered. (By Lemma 7.)

b) If a critical point (CPn) is inserted between two actual or virtual critical points (CPa and CPb), then a new arc will be appended to the LDTS. The arc will extend from CPn on C to a new vertex on the LDTS lying between the two vertices which have CPa and CPb as nn on C. The geometry of the LDTS will remain unaltered. (By Lemmas 5 and 6.)

c) The movement of a critical point can be considered to be a deletion followed by an insertion, and thus the resulting changes in the LDTS are specified by (a) and (b) of this theorem.

We will invoke the following results from Blum [1967, 1978] and Bookstein [1979]:

B1) Let S2 be the set of points of the (formal) skeleton S with exactly two nearest neighbors in the contour set C. Then S is enlarged from S2 by the branch points and end points of the open arcs of S2.

B2) Let C2 be the set of points of C which are nn in C of the points in S2. The set C2 differs from C by deletion of of some isolated points and circular arcs.

B3) If p1 and p2 are the two nn in C2 of q in S2, and if F(p1,q) is the function that maps p1 into p2, then F "partitions" C2 into pairs of arcs (P1,P2) with possibly overlapping closure points, each such pair associated with an open arc Q of S2. We will call the triple (Q,P1,P2) an F-set.

B4) Lines connecting points on S2 to their nn on C2 do not cross.

Figures B-2 and B-3 show some examples of the relationship between the boundary of an object and its formal skeleton (FS).

LEMMA 8: An arc Q of S has a vertex of order one if and only if the two boundary arcs in its F-set are either adjacent on C, or are separated by a circular arc.

Proof (see Figure B4): Consider the wedge formed by connecting a point q, on an arc Q of S2 with a vertex of order one, to its two nn on C (p1 and p2). If there were some non-circular arc P' of C, located between p1 and p2, which was in the F-set of an arc Q' of S other than Q, then lines from points on Q' to their nn on P' would cross either (q,p1) or (q,p2) in violation of result B4. Thus, the two boundary arcs in the F-set containing Q must either be adjacent on C, or be separated by a circular arc (circular arcs of C can generate vertex points on S rather than arcs, and thus do not explicitly appear in any F-set).

Let us assume that we have two boundary arcs (P1 and P2), which are either adjacent on C or are separated by a single circular arc, and are in the same F-set as arc Q of S2. Consider the shaded area shown in Figure B4. Q must have a vertex somewhere in this shaded area; otherwise, it would have to cross (q,p1) or (q,p2) violating result B4; Similarly, if this vertex had order greater than one, then a line from this vertex to a nn on C would have to cross (q,p1) or (q,p2).

THEOREM 3:

If the critical points are chosen to be the common points of closure of the two boundary arcs (P1,P2) contained in each of those F-sets for which the boundary arcs are adjacent on C, and, additionally, for each circular arc on C which separates two boundary arcs in the same F-set, we chose any one of its points to be a critical point, then

a) the LDTS will subsume the formal skeleton (FS), and

b) the LDTS and the FS will have identical topologies.

In less formal terms, this theorem states that if critical points are located at those places on C at which the "free" ends of the FS would intersect C if extended (typically at the angle points and circular arcs of the contour, with some reasonable tolerance permitted), then the LDT will exactly reproduce the FS with the free ends extended out to the contour, and intersecting it at the critical points.

Proof (part a): The LDTS will subsume the FS if critical points are located at the points of closure of the boundary arcs in all F-sets associated with every arc of the FS; this placement of critical points will assure that each pair of boundary arcs in every F-set have distinct labels, and thus every pair of points (one from each of the pair of arcs in a given F-set), which are nn to some point on the FS, are also distinctly labeled nn for some point on the LDTS.

Let us now delete the critical point between any two boundary arcs that are not members of the same F-set. The resulting relabeling of C will be such that two adjacent arcs will now have the same label, but every pair of arcs in each F-set, with the exception noted below, will still be distinctly labeled. This process can be continued until we remain with only those critical points specified by the theorem, while sill ensuring that the LDTS still subsumes the FS. The only exception is the case of a circular boundary arc as shown in Figure B-2. A circular boundary arc can generate a single vertex point on the FS and thus need not be explicitly represented in any F-set, but can still separate the two boundary arcs associated with the branch (Q) of the FS terminating in this vertex (Lemma 8). By inserting a critical point anywhere on the circular arc (asthetically at its center), we prevent the two adjoining boundary arcs in the same F-set from assuming the same label (and thus prevent Q from being eliminated). QED

Proof (part b): In the above proof, we have shown that the LDTS subsumes the FS when critical points are located between boundary arcs associated with arcs of the FS having vertices of order one (see Lemma 8). Thus, the LDTS and the FS have the same number of arcs with vertices of order one (see Lemma 2). The LDTS cannot have any arcs beyond those contained in the FS, since any such additional arc would have to extend between two internal points of the FS, thus creating a loop in violation of Lemma 2 that asserts that the LDTS is a tree (under the originally given assumptions). Thus, we have established that the LDTS (with the indicated placement of critical points) will be identical to the FS, except that it can extend those arcs of the FS that terminate in vertices of order one. QED

If we remove the simplifying assumption of a closed, simply-connected object boundary, then it still appears that a proper placement of critical points will allow us to subsume the FS, with the possible addition of some extraneous arcs with vertices of order one (see Figure B-5). However, under these conditions many of the results proved in this appendix will no longer hold; for example, the LDTS need not be connected (see Figure B-5c).

THEOREM 4:

The LDTS can be directly derived from a discretely defined boundary C, of n points, in O(n log n) operations.

> Proof: An O(n log n) algorithm exists (Shamos and Hoey [1975]) to generate a Voronoi diagram on n points; the Voronoi diagram partitions the plane into n V-regions associated with the n boundary points. There cannot be more than a total of O(n log n) boundaries because the algorithm which generated these straight line boundaries required at least one operation per boundary generated; therefore, deleting the boundaries between commonly labeled V-regions so as to construct an LDT-partition requires at most O(n log n) operations. The boundaries of the LDT-partition define the LDTS. QED

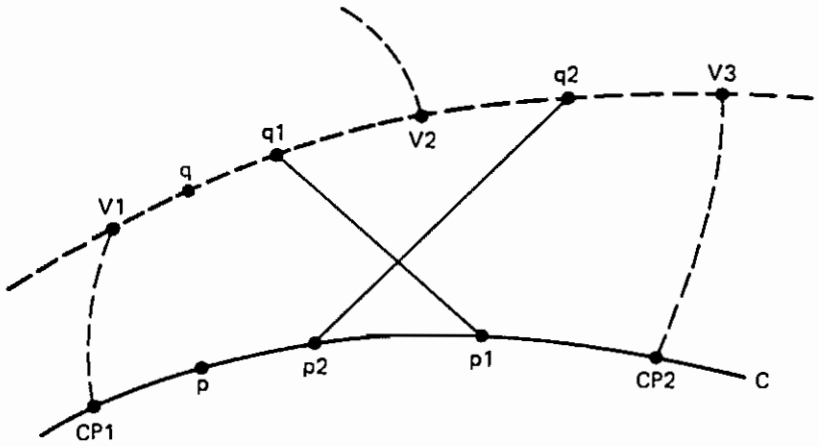


FIGURE B1: Properties of the LDT skeleton (see lemmas 4,5,6,7). The solid arc from CP1 to CP2 is part of the contour of C. The dashed path Q = {CP1,V1,V2,V3,CP2} is part of the LDTS.

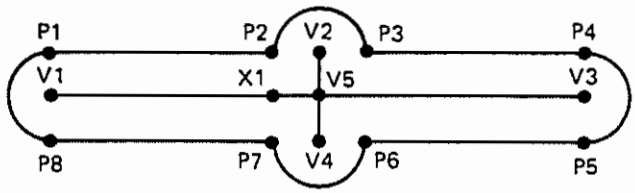


FIGURE B2: Examples of the relationship between the boundary of an object and its formal skeleton.

a) Circular boundary arc (P1,P8) corresponds to the single skeleton point (vertex) V1.
b) Skeletal segment (V2,V5) is generated by the the two boundary points P2 and P3.
c) Boundary point P2 is a nearest neighbor (nn) to the portion of skeletal arc (V1,V5) which extends from X1 to V5; P2 is one of the boundary points causing the formation of vertex V5, and is a nn to the entire skeletal segment (V2,V5).
d) The F-sets for this figure are:

F1 = { (V1,V5) (P1,P2] [P7,P8) }
F2 = { (V2,V5) [P2,P2] [P3,P3] }
F3 = { (V3,V5) [P3,P4) (P5,P6] }
F4 = ((V4,V5) [P6,P6] [P7,P7] }

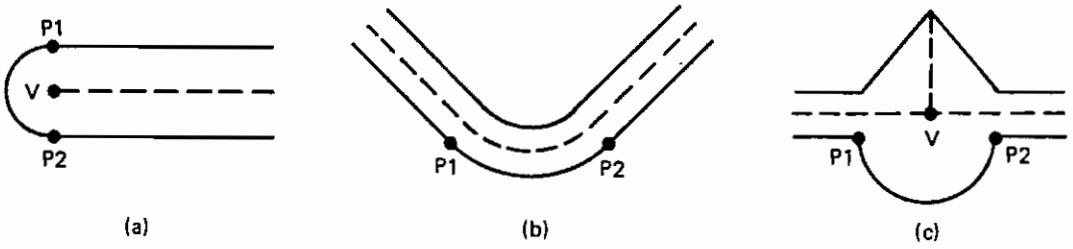


FIGURE B3: Examples of how a circular boundary arc (P1,P2) can affect the object skeleton.

a) (P1,P2) maps into the skeletal vertex V of order one.
b) (P1,P2) has center of curvature outside the object boundary, and maps into a segment of the skeleton without contributing to the formation of a vertex.
c) (P1,P2) maps into a skeletal vertex of order three.

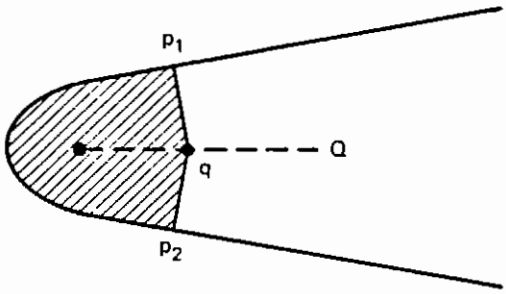


FIGURE B4: A property of the formal skeleton (see lemma 8).

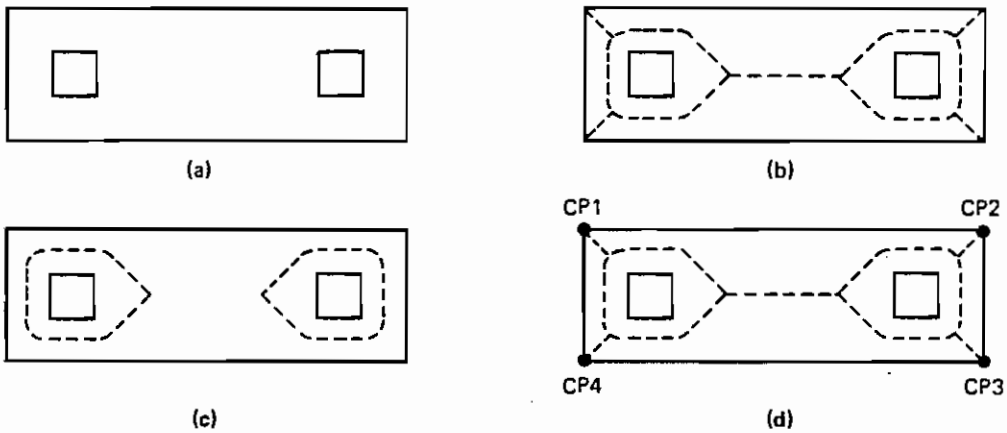


FIGURE B5: Example of the formal skeleton (FS) and LDT skeleton (LDTS) of an object with a non simply connected boundary.

a) Object with two internal boundaries.
b) Formal Skeleton.
c) Disconnected LDT skeleton (with no critical points).
d) LDTS is connected and subsumes the FS (given the indicated critical points).